What’s up with Random $k$-SAT

Dimitris Achlioptas

Microsoft
Given a Boolean formula (CNF), decide if a **satisfying** truth assignment exists.

\[
(\overline{x}_{12} \lor x_5) \land (x_{34} \lor \overline{x}_{21} \lor x_5 \lor \overline{x}_{27}) \land \cdots \land (x_{12}) \land (x_{21} \lor x_9 \lor \overline{x}_{13})
\]

**Cook’s Theorem**: Satisfiability is NP-complete.

---

**$k$-SAT**: Each clause has exactly $k$ literals.

---

$k = 2$: Pick any variable and set it arbitrarily. (1 choice)

- Satisfy any implications (repeatedly).

- Either get a subformula or a contradiction.

$k \geq 3$: NP-complete
Since the mid-70s a number of models have been proposed for Random SATisfiability.

Most models generate formulas that are too easy.

Let $A_{k,n}$ be the set of all $2^k \binom{n}{k}$ $k$-clauses on $n$ variables. [with distinct, non-complementary literals]

$\mathcal{F}_k(n, m)$: a random $k$-SAT formula with $m$ clauses over $n$ variables, formed by selecting uniformly at random $m$ clauses from $A_{k,n}$ [with replacement]

For all $k \geq 3$ and $r > 2^k$, there exists $\rho(k, r) > 0$ such that almost surely: $\mathcal{F}_k(n, rn)$ is unsatisfiable but every resolution proof of its unsatisfiability has at least $2^{\rho m}$ clauses.

[Chvátal, Szemerédi 88]
Satisfiability Threshold Conjecture

Conjecture: For each $k$, there exists a constant $r_k$ such that for any $\epsilon > 0$,

$$
\lim_{n \to \infty} \Pr[\mathcal{F}_k(n, m) \text{ is satisfiable}] = \begin{cases} 
1 & \text{if } m = (r_k - \epsilon)n \\
0 & \text{if } m = (r_k + \epsilon)n
\end{cases}
$$
Known Results

\[ k = 2 : \text{ Yes, } r_2 = 1. \] [Chvátal, Reed 92], [Goerdt 92], [Fernandez de la Vega 92]

Idea: Look at the “forced choices” branching process.

\[ k \geq 3 : \text{ We don’t know if } r_k \text{ exists.} \]

Easy bounds:

\[ \frac{2^k}{k} < r_k < 2^k. \]

[Friedgut 97]: For each \( k \geq 2 \) there exists a function \( r_k(n) \) such that

\[
\lim_{n \to \infty} \Pr[\mathcal{F}_k(n,m) \text{ is satisfiable}] = \begin{cases} 
1 & \text{if } m = (r_k(n) - \epsilon)n \\
0 & \text{if } m = (r_k(n) + \epsilon)n
\end{cases}
\]

Idea: All small subformulas are innocuous.
Upper bounds come from probabilistic counting arguments.

Pure literal heuristic: satisfy only literals whose complement does not appear in the formula. Exact analysis gives $\gamma_3 = 1.637...$
If there exist 1-clauses (unit clauses)

then

pick a 1-clause u.a.r. and satisfy it

else

select a literal \( \ell \) and satisfy it

• Value assignments are permanent (no backtracking)
• Failure occurs iff a 0-clause is ever generated
• The algorithm goes on to set all the variables even if a 0-clause is generated

select

UC: Pick a variable \( x \) u.a.r.; select \( \ell \in \{x, \overline{x}\} \) u.a.r. 8/3
UCwm: Pick a variable \( x \) u.a.r.; select \( \ell \in \{x, \overline{x}\} \) that appears among more 3-clauses. 2.9
GUC: Pick a shortest clause \( c = \ell_1 \lor \cdots \lor \ell_q \) u.a.r.; select \( \ell \in \{\ell_1, \ldots, \ell_q\} \) u.a.r. 3.003
For all $0 \leq i \leq 3$ and all $0 \leq t \leq n$: 

The set of $i$-clauses remaining after $t$ steps is uniformly random conditional on its size.

- Initially, all cards are “face down”; 3 cards per clause.

- We can
  \[
  \begin{cases} 
  \text{Name a variable} & \\
  \text{or} & \\
  \text{Point to a card} & 
  \end{cases}
  \]

- As a result, all cards with the named/underlying variable turn “face up”.

- After we set the variable: all cards corresponding to satisfied clauses get removed; all cards corresponding to the unsatisfied literal get removed.
$C_i(t)$ is the number of $i$-clauses remaining after $t$ variables are set.

- If for some $t$, \( \frac{C_2(t)}{n - t} > (1 + \delta) \) the algorithm will a.s. fail.

- The expected number of 1-clauses generated in round $t$ is $\frac{C_2(t)}{n - t} + o(1)$.

- If for all $t$, \( \frac{C_2(t)}{n - t} < (1 - \delta) \) the algorithm succeeds with probability at least $\psi = \psi(\delta) > 0$.

If for some $r^*$ we can show that a.s. \( \frac{C_2(t)}{n - t} < (1 - \delta) \) for all $t$, then $r_3 \geq r^*$. 
Tracing the number of 2-clauses

Unit Clause

Every step is "the same": we assign a random value, to a randomly chosen unset variable. Implication: $C_2(t)$ is a sum of Binomial random variables. Maximizing over $t$ gives

$$r_k \geq \frac{2^k}{k}.$$ 

Alternatively: $\mathbb{E}(\Delta C_2(t)) \equiv \mathbb{E}(C_2(t + 1) - C_2(t)) = f(t, C_2(t), C_3(t))$.

All Other Algorithms

- There are "forced" steps (1-clauses are present) and "free" steps (1-clauses are not present).

$\mathbb{E}(\Delta C_2(t))$ depends on $t, C_2(t), C_3(t)$ and on whether $C_1(t) \neq 0$.

$$\mathbb{E}(\Delta C_2(t)) = f(t, C_1(t), C_2(t), C_3(t)).$$

Knowing each of $t, C_1(t), C_2(t), C_3(t)$ within $o(n)$ is not good enough.
A “Workaholic” Server Lemma

A simple queueing system

- Time is discrete $t = 0, 1, 2, \ldots$

- In each step, Poisson($\lambda$) packets arrive at the queue.

- In each step, the server serves precisely one packet.
  
  (If packets exist; otherwise, the server goes to sleep.)

Queueing Theory:

If $1 > \lambda$, the expected queue size is bounded and the server sleeps $1 - \lambda$ fraction of the time.
Same queueing, different server

- Time is discrete $t = 0, 1, 2, \ldots$
- In each step, Poisson($\lambda$) packets arrive at the queue.
- In each step, the server flips a coin that says "wake" with probability $w$ and "sleep" otherwise.
  If awake, the server attempts to serve precisely one packet.

A 4000 thousand year tradition (with a recent proof): If $w > \lambda$, the expected queue size is bounded and the server sleeps $1 - w$ fraction of the time.

If there exist 1-clauses then pick a 1-clause u.a.r. and satisfy it. . .

vs.

With probability $(1 + \epsilon) \frac{C_2(t)}{n - t}$ attempt to pick a 1-clause u.a.r. and satisfy it. . .

Now $\mathbb{E}(\Delta C_2(t)) = f(t, C_2(t), C_3(t))$. 
[Kurtz 78, Karp Sipser 81, Wormald 95]

If we have random variables $Y_1, Y_2, \ldots, Y_k$ evolving jointly such that:

- At each step $t$,
  \[ \mathbb{E} [\Delta Y_i \mid \mathcal{H}] = f_i(Y_1/n, \ldots, Y_k/n, t/n) + o(1) \]
  where the $f_i$ are all Lipschitz continuous.

- The r.v. $\Delta Y_i$ have reasonable tail behavior.

Then w.h.p. $Y_i(t) = y_i(t) \cdot n + o(n)$ where $y_i(t)$ is the solution of $\frac{dy_i}{dt} = f_i$.

The evolution is stable under small perturbations of the state.
Differential Equations in action

\[ \text{UC} \]
\[
\begin{align*}
\mathbf{E}(\Delta C_3(t)) &= -\frac{3C_3(t)}{n-t} \\
C_3(0) &= r n \\
\end{align*}
\]
\[
\begin{align*}
s_3'(x) &= -\frac{3s_3(x) \cdot n}{(1-x) \cdot n} \\
s_3(0) &= r \\
\end{align*}
\]
\[
\begin{align*}
\mathbf{E}(\Delta C_2(t)) &= \frac{1}{2} \times \frac{3C_3(t)}{n-t} - \frac{2C_2(t)}{n-t} \\
C_2(0) &= 0 \\
\end{align*}
\]
\[
\begin{align*}
s_2'(x) &= \frac{3s_3(x)}{2(1-x)} - \frac{2s_2(x)}{1-x} \\
s_2(0) &= 0 \\
\end{align*}
\]

\[ \text{GUC} \]
\[
\begin{align*}
\mathbf{E}(\Delta C_2(t)) &= \frac{3C_3(t)}{2(n-t)} - \frac{2C_2(t)}{n-t} - \left(1 - \frac{C_2(t)}{n-t}\right) \\
C_2(0) &= 0 \\
\end{align*}
\]
\[
\begin{align*}
s_2'(x) &= \frac{3s_3(x)}{2(1-x)} - \frac{s_2(x)}{(1-x)} - 1 \\
s_2(0) &= 0 \\
\end{align*}
\]

\[
\frac{C_2(t)}{n-t} < 1 \iff \frac{s_2(x) \cdot n}{(1-x) \cdot n} < 1 \iff \begin{cases} 
\frac{3}{2} rx(1 - x) < 1 \iff r < 8/3 \\
\frac{3}{2} rx(1 - x/2) + \ln(1 - x) < 1 \iff r < 3.003\ldots 
\end{cases}
\]

UC

GUC
What is an algorithm to do?

- We want to minimize the \((3 \rightarrow 2)\)-flow to keep the 2-clause density low.
- We want to minimize the \((2 \rightarrow 1)\)-flow to have more free steps (opportunities).

If we can set \(\nu\) so as to minimize both flows simultaneously, then the choice is clear. But what if this is not possible?

Should one commit to what appears good now or sacrifice current benefits for greater flexibility in the future?

**Answer:** LP-duality.
Consider $\epsilon n$ rounds, where $\epsilon \to 0$.

We will encounter each 4-tuple $T$ of potential flows, very close to $\theta_T \times n$ times.

- The flattest possible slope at each point is given by a max-density knapsack problem.

Say $v$’s appearances are $(a_2^+, a_2^-, a_3^+, a_3^-)$. We set $v$ to 1 iff: $a_3^+ - a_3^- \geq \lambda(a_2^- - a_2^+)$. Moreover,

$$\lambda = \lambda(\rho_2, \rho_3) = \frac{s^*_2 - (\frac{3}{2}\rho_3 - 2\rho_2)}{1 - \rho_2}$$

This gives $r_3 > 3.22$ for one-at-a-time and $r_3 > 3.26$ for two-at-a-time.

[Ach., Sorkin 00]
Getting better algorithms

- Use a model for the analysis that allows explicit access to degree information: formulas are now uniformly random, conditional on their entire degree sequence.

- Dispense with "uniform-randomness" for the 2-clauses. Since 2-SAT is tractable, we can afford a less naive approach for 2-clauses.

<table>
<thead>
<tr>
<th></th>
<th>General $k$:</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC:</td>
<td>$\frac{2^k}{k}$</td>
</tr>
<tr>
<td>SC:</td>
<td>$1.12 \cdot \frac{2^k}{k}$</td>
</tr>
<tr>
<td>GUC:</td>
<td>$1.87 \cdot \frac{2^k}{k}$</td>
</tr>
</tbody>
</table>
First moment method

For any non-negative, integer-valued random variable $X$,
\[ \Pr[X > 0] \leq \mathbb{E}[X]. \]

Let $X$ be the number of satisfying truth assignments of $F_k(n, m = rn)$. Fix a t.a. $A$
\[ \Pr[A \text{ is satisfying}] = \left( 1 - \frac{1}{2^k} \right)^m. \]

Therefore,
\[ \mathbb{E}[X] = 2^n \times \left( 1 - \frac{1}{2^k} \right)^{rn} = \left( 2 \left( 1 - \frac{1}{2^k} \right)^r \right)^n = o(1) \]
if $2 \left( 1 - \frac{1}{2^k} \right)^r < 1$. Thus,
\[ r_k < 2^k \ln 2. \]

For general $k$, local maximality only gives
\[ 2^k \ln 2 - O(1). \]
Second moment method

For any non-negative random variable $X$, 

$$\Pr[X > 0] \geq \frac{\mathbb{E}[X]^2}{\mathbb{E}[X^2]}.$$  

Let $X$ be the number of satisfying truth assignments of $F_k(n, m = rn)$.

$$\mathbb{E}[X^2] = \mathbb{E}[(I_1 + \cdots + I_{2n})^2]$$

$$= \sum_{s,t} \Pr[\text{Both } I_s, I_t \text{ are satisfying t.a.}]$$

$$= 2^n \sum_{z=0}^{n} \binom{n}{z} \Pr[\text{Both } 00\cdots0 \text{ and } 00\cdots011\cdots1 \text{ are satisfying}]$$

$$= C \times \left(2 \max_{\alpha \in [0,1]} \left\{ \frac{(1 - 2^{1-k} + \alpha^k)^r}{\alpha^\alpha (1 - \alpha)^{1-\alpha}} \right\} \right)^n (\alpha \equiv z/n)$$

$$= C \times f(\alpha)^n,$$

where $f(1/2) = O(\mathbb{E}[X]^2)$. Unfortunately, $f$ is always maximized for some $\alpha > 1/2$...
Abstinence

Not-All-Equal Satisfiability: each clause must have at least one satisfied literal and at least one unsatisfied literal.

Observation: If $\mathcal{A}$ is NAE-satisfying, so is its complement $\overline{\mathcal{A}}$.

\[ X \text{ be the number of NAE-satisfying truth assignments of } F_k(n, m = r n). \]

\[ \mathbf{E}[X^2] = \ldots \]
\[ \leq \ldots \]
\[ = \ldots \]
\[ \leq C \times \mathbf{E}[X]^2, \]

for some constant $C > 0$. The existence of a sharp threshold completes the proof.

Intuition: NAE-assignments look like a “mist” on $\{0, 1\}^n$. This greatly reduces variance.
Random NAE-$k$-SAT

[First moment method] For all $\epsilon > 0$ and all $k \geq k_0(\epsilon)$, if

$$r \geq \frac{2^k \ln 2}{2} - \frac{\ln 2}{2} - \frac{1}{4} + \epsilon$$

then w.h.p. $F_k(n, rn)$ has no NAE-satisfying truth assignments.

Theorem 1 For all $\epsilon > 0$ and all $k \geq k_0(\epsilon)$, if

$$r \leq \frac{2^k \ln 2}{2} - \frac{\ln 2}{2} - \frac{1}{2} - \epsilon$$

then w.h.p. $F_k(n, rn)$ has exponentially many NAE-satisfying truth assignments.

[Ach., Moore STOC ’02]

<table>
<thead>
<tr>
<th>$k$</th>
<th>3</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our bound</td>
<td>3/2</td>
<td>9.973</td>
<td>21.190</td>
<td>43.432</td>
<td>87.827</td>
<td>176.570</td>
<td>354.027</td>
<td>708.925</td>
<td>1418.712</td>
</tr>
<tr>
<td>Upper bound</td>
<td>2.214</td>
<td>10.505</td>
<td>21.590</td>
<td>43.768</td>
<td>88.128</td>
<td>176.850</td>
<td>354.295</td>
<td>709.186</td>
<td>1418.969</td>
</tr>
</tbody>
</table>
Observation 1: The “populist” t.a. have a huge advantage towards satisfiability.

Observation 2: This advantage disappears for NAE $k$-SAT.

Let $X$ be the number of satisfying truth assignments of $F_k(n, m = rn)$ that satisfy

$$\frac{mk}{2} + O(\sqrt{m})$$

literal occurrences, i.e. as many as a random assignment.

**Theorem 2** For all $\epsilon > 0$ and all $k \geq k_0(\epsilon)$,

$$r_k \geq 2^k \ln 2 - k(\ln 2)/2 - 1 - \epsilon .$$

[Ach., Peres 02]